A circular emblem with a shield and text

AI-generated content may be incorrect.

CS-221: Data Structures and Algorithms

Semester Project

**Deliverable #1:** Project Proposal & Conceptual Design (07/10/25)

**Course Instructor:** Dr. Zubair Ahmad

**Group Members:**

1. Abdul Raffay Gill (2024019)
2. Syeda Maheen Kazmi (2024627)
3. Youraj Ali Karim (2024666)

**Repository Link:** [**https://github.com/syedamaheenkazmi172/Blockchain\_implementation\_using\_DSA.git**](https://github.com/syedamaheenkazmi172/Blockchain_implementation_using_DSA.git)

Mini Implementation of Blockchain using Data Structures and Algorithms

1. **Introduction:**

Blockchain is more than just modern technology. The name itself has become synonymous with transparency, security and privacy. Blockchain is a decentralized digital database/ledger that securely stores records across a network of computers in a way that is transparent and unchangeable. Each "block" contains data, and the blocks are linked in a chronological "chain." Blockchain is at the core of all cryptocurrencies and is being integrated into banking, supply chain, healthcare and almost all major industries.

Despite the ever-increasing usage of Blockchain globally, the average person has little to no idea of its working and its processes. As such, for our CS-221 Semester Project, we will be working on small scale replica of Blockchain using different concepts we learn in our “Data Structures and Algorithms” course.

Our project aims to make Blockchain accessible and easy to understand for the masses and to provide them with a platform to interact with and learn the processes behind Blockchain. Through our project, we aim to increase technical literacy and demystify Blockchain.

1. **Key Data Structures**

* Doubly Linked List (DLL)
* Use: Chain of blocks (backbone).
* Why: Each block links to previous & next, so we can traverse forward/backward.
* Queue (FIFO)
* Use: Pending transactions (mempool).
* Why: Transactions arrive in order and wait for confirmation.
* Priority Queue (Heap)
* Use: Transaction selection by priority (e.g., higher fee = higher priority).
* Why: In real blockchains, miners pick high-fee transactions first.
* Hashing (Functions / Hash Maps):
* Use: Create block fingerprints + validate data.
* Why: Guarantees immutability
* Graph
* Use: Network of nodes.
* Why: To simulate multiple computers validating and broadcasting.
* Binary Tree / Merkle Tree
* Use: Store transaction hashes inside a block.
* Why: Instead of checking all transactions, you only check the root hash.

1. **Main Algorithms (**To Be Decided)
2. **Data Flow**

The following is the projected flow of data; from being received, to being processed and displayed.

===========================  
 BLOCKCHAIN MINI SYSTEM  
===========================  
1. Create Transaction  
2. Mine Pending Transactions  
3. View Blockchain  
4. Check Balances  
5. Exit

* If User Chooses 1 (Create Transaction)  
  Enter Sender: Alice  
  Enter Receiver: Bob  
  Enter Amount: 5  
  Transaction added to Pending Pool!
* If User Chooses 2 (Mine Pending Transactions)  
  Mining started...  
  Proof of Work found! Nonce = 15402  
  Block #1 created with 3 transactions  
  Miner rewarded 10 coins  
  Block added to Blockchain
* If User Chooses 3 (View Blockchain)  
  Block #1 [Hash: 0000abcd...]  
  Previous: 0000xyz...  
  Transactions (Merkle Root: abcd1234...):  
   Alice -> Bob: 5  
   Miner Reward -> Miner: 10
* If User Chooses 4 (Check Balances)  
  Balances:  
  Alice: 95  
  Bob: 5  
  Miner: 10

1. **Integrating the Course:**

Throughout the working of this project, it will be crucial to understand the concepts being taught in class. Course concepts like the different ADTs will be the focus in the working of this project. To keep our project efficient, we will be conducting space-time analysis, which is yet another part of the course. Every lecture will provide a new idea that we could potentially integrate within our project.